ICANN78 | AGM – Planning for String Similarity Review in the next gTLD Round
Thursday, October 26, 2023 – 9:00 to 10:00 HAM

PITINAN KOOARMORNPATANA: Thank you. Hello, and welcome to the planning for spring similarity review in the next gTLD round session. My name is Pitinan and I am a participation manager for this session. Please note that this session is being recorded and is governed by the ICANN Expected Standards of Behavior. This session includes interpretation and real time transcription, which can be enabled by clicking on the icons in the Zoom toolbar. If you would like to speak, please raise your hand and the moderator will call upon you. Alternately, you may type your questions in the chat. All participants are kindly asked to state their names before recording and speak at a reasonable pace. With that, I'll hand over the floor to you, Sarmad. Thank you.

SARMAD HUSSAIN: Thank you, Pitinan, and good morning everyone here and good morning, good afternoon, good evening to people who are joining us online. This particular session is possibly the first in a series of sessions, which will be focused on how we are planning for string similarity review in the next gTLD round. And I'm joined here by Asmus Freytag, who'll be talking more about some of the guidelines we've been developing along with Michelle DeSmyter, they're consulting with us on this project. Before we get into the overview of the presentation itself, I thought it may be useful to set a bit of context of what we mean by

same, similar and distinct because those are some of the things we'll be talking about during our conversation today.

So when we are talking about strings or TLD labels, which are applied for or are being used, they are of course composed of characters or code points in Unicode and they can visually speaking, they can be same or similar or distinct from each other. And when the code points are combined together or these characters are combined together to form larger strings, the corresponding strings then also can potentially be considered visually same or similar or distinct. So the similarity can be at two layers, the character itself, which compose the string and at the string level and in some of the scripts, interestingly, when you combine characters in strings, they actually change their shape.

So just looking at individual characters is not sufficient, though it does provide some information, but eventually for string similarity review, one would also need to look at the whole string as well. But just to sort of set the context, here are some examples of characters and their Unicode code points from some different scripts to show you some of the categories. The first row here, which is visually the same, basically represents those characters which the users of that script or community which uses that script consider either same or indistinguishable from each other. In some cases, they may be slightly different, but not discernible easily. And in some cases, they are actually exactly the same, meaning they're actually in the font being used as using exactly the same glyph or image to encode that character.

And if you look at the Latin and Cyrillic example, the Latin character C and the Cyrillic character which has a code point of 0441, they're exactly

the same. Similarly the Arabic, and so this similarity can be across scripts, but it can also be within a script. So if you look at the Arabic example, you will see that the character which is represented or encoded in Unicode at 0649 and 06CC, they're also identical. At least in this context. So those are examples of what we call visually same.

So string similarity process, we are actually only looking at visual interpretation and not any other interpretation. Visually similar characters, these are identified as such by the script communities which I've been working on the root zone label generation rule. So basically gives you an example of two characters from Latin and Greek, which may be considered similar.

And then one from Devanagari and Bangla, which may actually be considered similar. And then other characters, the last row shows things, characters which are totally distinct from each other. For example, A and F in English or just arbitrary other characters in other scripts and languages. So the first row which represents characters which are same or indistinguishable have already been identified by the script communities and they've actually been already tagged explicitly as variants in the root zone label generation rules. So that part is, I think, already encoded and already addressed.

What to this part of the work, which string similarity review is concerned, is mostly concerned with this middle row where there are characters which are visually similar and then possibly strings formed with those characters which are visually similar. And then it is then up to the independent review panel which will be set up for the next round,

which will then determine if those, if two strings fall in this category that they are visually similar, whether they are confusingly similar or not.

If they are confusingly similar, then they should not proceed because that creates confusion in the top-level domains which are delegated and that's not a good thing for end users. But if they are similar but still distinguishable, then they may proceed. And that's sort of a finer distinction which needs to be made by the string similarity review panel and that's what the string similarity review process is about. So what we are eventually working on is this middle row and trying to tease out details to see what is a good mechanism to make that process more robust.

So let's get into the presentation itself. String, what we'll do in this session is before we actually get into more details, we'll just overview some background recommendations coming from policy which is motivating this work. The first set of recommendations come from the final report on the new gTLD subsequent procedures policy development process or SubPro for short. And then in addition, there are more details which are added for string similarity. In the Phase 1 recommendations of the internationalized domain names expedited policy development process or IDN EPDP for short. Please note that IDN EPDP is currently published. It's phase one recommendations for public comment. And what we are sharing here are the recommendations which have been published during the public comment.

So these are not the final recommendations from EPDP. They have taken input from through the public comment process and they are

now revised some of those recommendations. But those final recommendations are not yet published. So some of the details we're sharing on IDN EPDP may change once the final recommendations are published. But for this presentation's purpose, we are using the version which was actually published for the public comment version until the next one becomes available. So with that, we will then review those recommendations and propose or share based on those recommendations the scope tentative methodology and outcomes of the string similarity review for the next round gTLD round.

And then one of the as you will see these recommendations call for string similarity review guidelines to be developed for implementation through the string similarity review panel which will be doing this work. So we've invited Asmus who's helping us to develop those guidelines which will eventually feed into this process. These guidelines are not currently final there we are just starting to develop these guidelines. So what we'll do is Asmus will be sharing the current thinking process where we are going with this and we would obviously seek your input here to see whether this is the right direction to go in what other considerations we may want to take in the process of developing these guidelines. So that's sort of an overview of the session and let's just get right into it.

So starting from the policy recommendations basically section 24 of SubPro talks or focuses on string similarity review and the working group of course affirms recommendation 2 from the 2007 policy which states that strings must not be confusingly similar to an existing top level domain or a reserve name. So that's sort of the baseline of where we're starting from. Affirmation 24.2 so this is also building up on the

2012 round of gTLD which say gives some more detail on what are the scope of those string review comparisons but more importantly it gives us a bit of a definition of what confusingly similar means for strings and if you read it says similar means strings so similar that they create a probability of user confusion if more than one of the strings is delegated in the root zone.

So basically the string similarity panel was eventually tasked for determining what differentiates between just a simple similarity versus that similarity which will be confusing for the end users with some probability and that sort of the details which will eventually need to be teased out. There are some additional recommendations which talk about automated tool which may not be sufficient to do this and manual processing is needed some details on objection process.

And then in the IDN EPDP Phase 1 recommendations there's much more detail which is added for string similarity review. Recommendations 4.1 clearly delineates the scope of the comparisons of the applied for string must be compared with existing gTLDs, existing ccTLDs, strings requested as IDN ccTLDs, other applied for gTLDs, all strings on the reserve names list and any two character ASCII which are reserved for country codes.

And it actually not only lists these but suggests that not just the original string and the string these strings will be compared but their allocatable variants with be compared with allocatable variants and block variants. And then the block variants also will be compared with other block variants. And we'll go into that detail later as well, but it actually gives

a lot of detail of what those comparisons may actually look like. So that talks about the scope of comparisons in that recommendation.

Then it also actually goes into saying that it realizes that sometimes that scope of comparison may become very large, because if you have comparing strings and their variants with other strings and their variants, sometimes those sets become, a string may actually have a hundred different variants. And so you're comparing a hundred variants with a hundred variants of the string and of existing strings.

So in case where some of these comparisons may become large and it may be very clear that some of those comparisons may not actually lead to any string similarity issues, then the string similarity panel could actually omit some of those comparisons. But those, so it allows some efficiency in the process, but only to the extent that, if you're for example, comparing Chinese with Arabic, maybe the likelihood of similarity is very low between those script pairs.

But if you're comparing Cyrillic and Latin, the likelihood of similarity may be very high. So it shouldn't be just arbitrarily reduced and some very detailed guidelines should be developed on what can be, well, it already says what should be compared, but those guidelines and criteria may then define what could be potentially skipped where there is very low chances of similarity across different pairs of scripts. So these particular guidelines allow for those exceptions, but through a guideline, which of course has to be developed beforehand.

And finally, this actually recommendation says that if you actually have a similarity issue with any one of the strings in a variant set, then the whole variant set gets impacted, it's not just that string. So if one string

is found similar to another string in applied for next gTLD round, for example, then the entire variant of that string and its variant set will go into for example, contention with the entire variant set of the other string, not just those two strings, for example. And also, of course, if variants match directly, even if they're not similar, they also go into the contention set. So that's sort of the context of policy, guiding policy, which is asking us to develop the scope methodology and outcomes. So coming back to the scope, as you can then see, what is being suggested is on the top here is the applied for gTLD string.

And it is the primary string itself, the string which is applied for, and then it can have some allocatable variant labels and it will have block variant labels. Those are calculated through the root zone label generation rules. Similarly, on the other hand, on the left-hand side, you'll see the categories of comparison which have been identified in the ID and EPDP. These include the existing gTLDs or gTLDs, for example, which have been applied for, but are still in the process from the previous round.

So there are still strings which may be not processed, for example, from the 2012 gTLD round. So those are also a part of this process. Existing ccTLDs or country code top-level domains and other country code top-level domains which are in the process of application. So they're requested, but not kindly delegated or successfully evaluated. Other applied for gTLD strings in that same round, reserved names and any two-character ASCII because two-character ASCII are reserved for country code top-level domains managed by ISO 3166 management agency.

So basically, they can also have the primary string, allocatable variants and block variants as calculated through the root zone label generation rules. And all those strings need to be compared with all the applied for string and their variant labels will need to be compared with all of them. The yes basically marks where the comparisons are required. The yes with asterisk or those areas which are slightly grayed out, list those comparisons which the panel may decide to omit in case, for example, as I was sharing earlier, if you're comparing Chinese and Arabic, the panel may decide that the chances of similarity are very low and therefore it can potentially for some particular string decide to omit those comparisons. Those are the ones which are marked as yes asterisk. So as you can see, panels still cannot omit all the comparisons. There are only some comparisons which it can omit.

And then no marks the area which are not in the scope of comparison for string similarity review. So the block variants versus block variants are not going to be compared in the process. This is a summary of what is actually provided in the policy 4.1, the recommendation 4.1 from the IDN EPDP. Methodology of course is that we actually at the starting point, we actually have to compile a list of all these strings which need to be compared. They need to come from all the different categories. And we also need to compile a list of the allocatable variants and also possibly the block variants.

In some cases, the block variant list may be large for a particular string, not generally, but for a particular string, it could actually be large, in which case we will need to figure out a more feasible mechanism to do those kinds of comparisons. And that of course, detail will be left somewhat to the guidelines and some of it to the panel itself. Then once

the complete set of strings are identified, the panel will decide on which block strings to omit or which comparisons to omit, if any, and document the rationale for that decision. Then they will identify the different applications with the same variant string set for contention. So if there are strings which are variants of each other, two different applications, they should obviously go into contention.

So they will go into contention because of that. Or if there are two strings which are the same, they obviously go into contention as well. And then for the remaining, they will conduct the comparison of strings for string similarity to the extent that they're confusingly similar. And that definition of what is confusingly similar will obviously be coming from the string similarity review guidelines which are being developed. And then based on that, the panel will make contention sets identified, document the contention sets and the rationale of the string similarity review. And this will come back from the panel to ICANN and the results will be published. So that is sort of a very high level view of the process which will be followed and based on the guidelines which are provided to us, recommendations and guidelines.

Outcomes, of course, there are multiple possible outcomes. String review panel will document the following outcomes. There can be a string which is applied for, which is similar to an existing gTLD that needs to be documented. And in that case, of course, it may not be able to proceed, for example. We'll talk about that in the next slide. They can document that a string is similar to a gTLD string which is applied for in the previous round or similar to an existing ccTLD or similar to a requested ID and ccTLD or similar to a reserved name or similar to two character strings. Or what's not listed here is when this is similar to one

of these strings, it means actually either it is similar or it's allocatable variant is similar to, it's allocatable variant or block variant. So all the options we discussed in the previous slide, if any of those hold, this outcome will come out.

Or finally, they find that string is not similar to any of the categories which are listed in which case the string can obviously proceed. And I can publish the outcomes of the strings multi-review on its website for everyone to review as part of the process which is being developed. So this goes into a little more detail. I'm not going to go into and read through the whole thing, but those outcomes have eventually some consequences.

So just to take the simple example on the top left, if the first row, if the variant applied for gTLD or any of its variant strings is found to be same as an existing gTLD, it cannot proceed. If it is found a variant of an existing gTLD string, it cannot proceed until of course it's an allocatable variant and it's applied for the same registry operator in which case it's a variant application and then it can of course proceed.

Or if it's visually similar to, but not a variant of an existing gTLD, then it cannot proceed. And you can go through the slides and see the other set of consequences of the string similarity review panel outcome. And this of course will then guide us on developing the process itself. With that, I'm going to hand it over to Asmus who's going to take you through the guidelines themselves which are being currently in process of development and over to you Asmus.

ASMUS FREYTAG: Well, thank you, Sarmad. You have covered nearly everything I was going to say. So let's go to the next slide. And let's go to the next slide. And just to reiterate, confusability is the degree to which two labels can be substituted for each other. And substituted is the key concept here. And the confusable, we think of some kind of threshold that exists that is not really objectively and repeatably definable, but we have to pick one where we say the confusability has reached a level that it matters for the root sound. It matters for the root sound because confusability clearly exists on a smooth spectrum from very little to very much confusability.

Next slide, please. The string similarity is the subset of confusability which is based on similarity in appearance. And therefore, you can have the concept of confusable similarity, which is the degree of similarity that you need to make the labels confusable, that is to exceed that nevertheless defined threshold that you assume past which the confusability is not tolerable anymore.

Next, please. The key realization is that it is very difficult to draw up a very firm standard of confusability because what looks confusable is different for every user because, and also different to the context in which a label is presented. People familiar with a given script or language in a given script have gone through years and years and years of reading strings in their script and have a certain way of looking at them that may not be shared by people familiar with a different language. So one of the first things in creating a guideline you have to do is you have to define what your target user is.

Next slide, please. So the first cutoff we will suggest is that the user that we're aiming for is what we call reasonably attentive. That is, if you're not really looking at what you're clicking on, you will confuse anything with anything. If you are a typographer, you will make distinctions that ordinary users won't make. So we'll need to pick some common sense middle ground here.

The other requirement that would be reasonable is to say, when it comes to internet addresses and strings, we're really aiming at people who are familiar with the script or language in which the label is presented. And that, however, means we can't really rule out the situation where a user is looking for a label in his native script or native language, but is instead being presented with something that actually is using a script foreign to the user. And that happens to have a similarly looking string. And however, what we would exclude from our target user definition or the definition of a target context is the situation where a user is trying to decide between two labels in a script that the user can't really read.

If we are trying to figure out what is the confusable similarity between two Chinese strings as viewed by a user who is native in the Ethiopic, that is an unanswerable question because to them, every single Chinese label might look like every other Chinese label. We are also assuming that a significant portion of users are de facto familiar with the ASCII subset. And we can also assume for historical reasons that many of them are in fact looking to find the correct ASCII label to access a resource. So a definition of a target user has to cover the context where somebody native in a different script is looking for an ASCII label.

Finally, in here, it should be another bullet. We are excluding the context where a user is trying to distinguish between two labels that are visually present at the same time. That is a situation that is very commonly used for artificial tests and research studies of confusability, but the real life situation is the user has an idea of a name and then looks at a physical embodiment of something and has to decide whether that physical embodiment matches what they're looking for. You rarely have the case that you are presented with two labels side by side and have to pick one of them.

Next slide, please. The key concept here is that when you're trying to do a string similarity review, it's definitely a string-based thing because it matters in the end, not whether individual code points are similar, but whether the resulting string is similar or confusable. One of the things that leads to, even if you have confusable code points, if a label contains a unique substring that is a substring that is composed out of code points that don't have any similar equivalence anywhere else, then that should be enough to make the entire label unique because there's no way to construct something that can be confused with that label, even if some other parts of the label have confusable equivalence.

The other interesting aspect is that when people look at labels, they treat them like their normal reading task or word recognition. And we know that the word recognition is holistic, that people don't read words letter by letter, they read the whole word. And that means that if enough of the whole word matches an expectation, it can set up a context that subconsciously triggers the recognition by the user even though certain details are wrong. And so therefore the holistic reading can override some small differences in presentation.

That means that even though something should be a distinct it is not typical case of that is many English readers will simply not recognize that you have added a diacritic or a small mark above or below a letter to create a fake label. In their reading, such marks do not exist. They are not noticed. They are at best experienced like dirt on the screen and users are quite happy to click on the wrong label that just has an accent in some place where it doesn't belong.

So I think I covered the third bullet already. So let's go to the next slide. It's worth distinguishing similarity from the concept of variants. Next. So with variants, we define a label that in some sense of the word same is treated by a user as same as the original label. That means it either has the same appearance or so close to the same appearance that the user will take it as the same. But variants are not limited to visual similarity which is a very important concept. Very common case is for instance, the simplified versus traditional form of Chinese characters. These are ultimately considered the same character and users may not consciously treat them as different even though they're not visually similar. And in the Ethiopic, we have an interesting case that many letters produce the same sound and are therefore freely substituted by people writing the language.

Again, they don't look similar but they are labeled spelled with one set of letters are treated as the same as label spelled with a different set of letters. The variant relationship is an equivalence relation indicated by the same as and all the equivalence relations are transitive. That means any variant of a label is also a variant of all the other variants as well. However, similarity in the general sense means that there is some difference though it's small.

And because of the difference that is detectable even though it may be ignored by readers is that two labels can be similar to a third label but they may not be close enough in appearance to each other to be similar. So similarity and confusability are not transitive which makes remove some of the efficient tools we have in handling variant sets.

Next please. So there are some variants that are based on visual similarity and if so, they have either identical or very nearly identical appearance. One of the things that variants do not consider is other variants that are not very similar to not consider is uppercase based similarity. So ASCII labels can be present in lower or uppercase and even IDNs are accepted in uppercase by most user agents and therefore you can have labels that are present in uppercase and the user is required to figure out whether that's the label they want. And we did not in the root zone include those in the variants because of transitivity. I have an example to show how that gets you into trouble. However, for string similarity where we do not have the restriction of being transitive the confusability based on the underlying uppercase equivalence should be in scope.

Next slide please. So here's an example of how this plays out. You start with the letter H at the top left and it has an uppercase H we often put in the ASCII set and most users will not consciously track whether they've seen a label with an upper or lower case and in fact, for ASCII labels that both forms are equivalently valid. The uppercase H matches precisely in shape with the uppercase eta in Greek. And that would be normally a variant except we ignore uppercase for variant definitions but you have a near perfect similarity. So if you have a label that consists out of an uppercase eta, a user cannot tell whether it is Greek

or Latin. The lower case form of eta is a bit more distinct but it is still similar enough to the lower case N that there is a confusability. Not potentially because not everybody, if things are presented in the right context may note the absence or presence of that little extender on the leg on the right hand.

The lower case n obviously has an uppercase N in ASCII which happens to be precisely the same in appearance as the uppercase nu in Greek. And again, it's not a variant because we exclude that but if a label is present in uppercase for you to click on, you cannot tell whether you're looking at a Greek or a Latin label. The Greek nu has a lower case form that you see there with the code point 03BD and it in turn is fairly similar to a lower case v and enough that you may think that in some context of some labels, you have a definite confusability. Now the lower case v has an uppercase equivalent of an uppercase V and if you try to complete the circle you realize that the uppercase V and the lower case h are absolutely visually completely distinct. There's no chance of confusing the two of them.

So that's one of the reasons why we didn't try to treat these things on the variant level where every one of these would have to be considered a variant of all the other ones. And you would have had all the Ns and Hs and Vs in Latin be variants of each other which is not very practical. But for a string similarity, we will have to include these effects because otherwise you can happily create very confusable labels. In fact, labels that users cannot tell which script they're in. To take the case, you make a label HN and if it's in uppercase, you can't tell is it Greek or Latin.

Next slide, please. So this repeats to a little bit the large matrix of things that the Sarmad has presented. So we have to, because all variants are considered quite same as of a label, they become confusable with any delegated label or its variants. Sarmad excluded the case of comparing a blocked variant with another blocked variant. And I think that is from a technical point of view, actually a mistake because even blocked variants are treated by the user as something they would accept as the same as, and to have two things that are plausible substitutes excluded from comparison seems an oversight to me.

Let's do the next slide. There are some interesting interactions with the different scripts. And let's explore those in the next slides. Here's another case, this is our lower case, how different scripts interact. So at the top, you have on the left a Latin character, Aang, which most of us don't need but it is used in certain languages. And on the right hand, you have a Armenian character that you can argue is either similar or distinct. I have shown these two with a very small overlap to show that many people might consider these distinct because the legs curve in the opposite direction. However, if you add into the mix, the Greek Ada, it has a leg that isn't curved one way or the other. And suddenly you have a very large overlap in similarity.

So you may, depending on whether you come down on, whether the two in the top are similar enough to be confusable, if you decide they're not similar enough, you may allow them both to be present, but then they cannot be present at the same time as an Ada. Next slide, please. The concept of that similarity is in the eye of the beholder really, really matters when it comes to scripts that are of very different nature, because what readers do for each script, they look at particular features

to be able to tell words apart from each other while they will ignore other features that people unfamiliar with the script might think of are important.

And so that which type of graphic features will matter is different from script to script. However, for some scripts, we have some available data where people have already thought about what might be confusable, while for other scripts we don't. And then there are scripts where the issue is very difficult because they are scripts where adjacent characters will form something very similar to what we know as ligatures from the Latin script.

So where you have a fused form that may look very different from the individual letters. And I believe a lot of research will have to be done to understand the scope of what kind of fused forms are potentially confusable with other fused forms in the same script. It is a place where looking at the whole label becomes even more important than for many other scripts.

Next slide, please. So another concept for scripts is that some scripts are related to each other. They can have either similar features or they can be related because of historical developments or both. So for Latin Greek, Cyrillic and Armenian, there's a common origin to all these scripts. And the result of that is that there are many, many shapes that are in common across. We have seen some example between Latin and Greek already. For Chinese, Japanese and Korean, you have for those languages, you have overlapping repertoire. All of them make some or prominent use of the Han script.

And in general, even the parts that are not directly the Han script have similar topographical features. The neo-Brahmi scripts of India and nearby countries have a common origin. They have a similar structure. Many shapes are common. An interesting thing to note is that you may think it's unfortunate, but we have had a development where particularly the fonts used in user interfaces, for example, fonts like Segway UI are the ones that show the strongest tendency for harmonizing glyphs across scripts.

So often you find Greek presented in a slightly different style overall from Latin, or you find the historically common form for Armenian that looks very distinct. But if you go into those fonts, they have all chosen to use a set of shapes that's harmonized where the Greek looks as close to Latin as it can manage, and even the Armenian looks close to Latin as it can manage, which actually increases the confusability of labels.

Next slide, please. To give you an idea, the two scripts of India, Kannada, and Telugu have 34 basically identical shapes. For Latin and Cyrillic, the set of identical shapes is 28. And for another set of Indic scripts, Devanari and Gomuki, the set of identical shapes is 26. Those are really high overlaps, and it's really easy to create a label in one of the scripts that looks totally indistinguishable from a label in the other script. On the other hand, you have scripts that can be strongly related like Thai and Lao, but they have historically evolved into using fairly distinct font styles, and therefore it is not as easy to create and it's not so readily possible to present the same label, a label that you can't tell which script is in.

And as we discussed, Latin and Greek in particular have many uppercase forms that are shared, and they become confusable if the label is presented to the user, an uppercase of which we don't have control. And in particular, all the non-IDN labels can look like Greek uppercase.

Next slide, please. One very important thing to be aware of is that the number of defined variants for a label, block variants, may be unexpectedly large. We have seen examples of Arabic labels where you may have the number of blocked variants in the 10,000s, or if it's a longer label, even a million. That means that when Sarmad talked about create a list of all blocked variants, this is something that in the general or even the typical case is simply not possible.

So you have to use some different kind of technique, but enumeration is the kind of brute force approach that will only work for like two-letter labels and not much else. It's simply not possible to individually compare all variants against all other variants. So you have to do something different. And it might be possible to do a two-stage stage process where you find an automated way of eliminating from comparison all pairs of labels that are unlikely to be confusable based on that you have decided they have enough distinct features to make them not confusable. And then hopefully that reduces the set of reviewable labels so that you can do a focused review of them on an individual basis.

Next, please. So the key point in that reduction to labels would be to have some automated process that is data-driven that works basically fairly similar to the way we detect collisions among blocked variants.

Next slide, please. And the way we're currently thinking that might be a successful approach would be to basically do something similar to the variant mechanism where currently we have, for variants, we have a variant that's either blocked or not. So it's a yes, no. And to replace that by something where you define something like a confusability quote variant but give it different strengths from dissimilar to identical. And which would, because of the identical, would treat the confusables as super set of variants.

Now, I have mentioned before that the confusables are not transitive. This approach would pretend that confusables were transitive and therefore overproduce confusability. And that is on some level desired. So you run a tool based on this kind of data and you get handed something that is a super set of what could be possibly actual confusables. And you can think of this as a loose variants definition and they would include some that really shouldn't be considered confusable, but that would be for the detailed review in the second stage to exclude.

Next slide, please. For Latin Greek, Cyrillic, Armenian, we have some single code point data as byproduct of the roots on LGR development. And some of the registries have policy documents where they've identified single code point confusables. There is a data set available from Unicode, the confusables data. The problem with it is that it has not been consistently maintained. Its direction is being reviewed in the Unicode consortium and it is in need of vetting and alignment with the root zone definition of variants.

At the moment, there are discrepancies that are not helpful. However, no matter what data you're looking at, there is generally very, very poor data for any sequences. That is, are there multiple code points in a row that are confusable with a different sequence of code points?

For instance, scripts that write in clusters, like all the Neo-Brahmi scripts, you would have to compare cluster to cluster, so you have sequence to sequence. But there are even examples that are not limited to those scripts, like in even in ASCII, if you write.corn versus.com, you can fool a certain percentage of users.

Next, please. So even if you do pre-screening, you still have the task to get to yes-no decision of is it confusables? And we're talking about developing guidelines. We have to be very clear that this cannot be an algorithm. If we could define an algorithm, we could write a tool to do it. So there'll have to be a final decision left to the string similarity review panel where they have to make a judgment call. They will probably have to consult script experts because what matters is how native readers look at a label.

And we are thinking of ruling out the use of focus groups because that's an expensive process. And there isn't a good methodology that exists. And finally, we recognize the need that even if we have a pre-screening process, we may need to have the string similarity review panel be able to explicitly add additional reviewable candidates. And when they're done, they will have developed a contention set along the lines that Sarmad described earlier.

Next slide, please. So we are in the middle of writing an initial report and collecting something like an initial data set, this to be followed by

a public comment period for the initial report. And if the community finds the methodology proposed acceptable, then we would do a trial run by pre-screening of the pre-screening process on some sample labels and also the already delegated labels just to see whether they would show up as confusable and then refine the data sets as appropriate.

And then finally, we'd get community input on those initial test results and finalized methodology. Next, please. So do you may want to run questions or?

SARMAD HUSSAIN: Well, so we've run out of time for questions. If there is, maybe we can take one or two questions quickly. Otherwise, Pitinan shared an email address in the chat. Please send your questions to that email and we'll be sure to respond to you or feel free to come and talk to us. So, Anil, let's go to you for questions, please. There's a mic on the right-hand side in case you want to come if you're here and you want to speak, thank you. And we can't hear you.

ANIL KUMAR JAIN: Anil, for the record, my question is that, suppose the string similarity review panel has seen the applied for script and domain and they approved it. But maybe when it start getting actual use in the system and we find that it is similar to either existing script or any script and there are complaints, what is the procedure at that particular stage either to hold back or to take some action? So what are the systems which are available with us as on it? Thank you.

SARMAD HUSSAIN: First of all, that should probably not happen. And eventually it would really be depend, string similarity review panel, there are additional checks and balances in the system. For example, there's a whole set of objection procedures which can be used as well by the community at large beyond the panel to address some issues which seem apparent, but if for some reason not captured by the string similarity review panel. So there are other checks and balances in the application process which actually could be used as well, I hope. But once it's then delegated, then obviously it moves forward, thank you.

ANIL KUMAR JAIN: Thank you.

SARMAD HUSSAIN: We'll just take one last comment and we have to close here over time.

NABIL BENAMAR: Thank you. This is Nabil for the record. As far as I know, we have already done this work for the Arabic script. It was about the variant, but for the similarity now or confusability, there is something new. we have within the same script, we can have two words that look similar, but they are written in two different languages. So I would like to ask if we have in this panel a subgroup working mainly on the Arabic script, because it has some specificities maybe that we don't find in other scripts. Thank you.

SARMAD HUSSAIN: So we'll, as Asma shared, we'll need to eventually develop some data sets beyond just the sets of variants which have been developed to look at similarity. Some root zone LGR panels have already developed some confusability data sets as part of their work. Some have not. So those are things we are obviously looking in and where those data sets are needed, we will of course work with the community to develop them. With that, sorry, we didn't leave much time for questions and answers, but please reach out to us through the email provided and let me thank all of you for attending the session. Thank you Asma for presenting and we'll close the session. Please stop the recording.

**[END OF TRANSCRIPTION]**